

PATENT ABSTRACTS OF JAPAN

(8)

(11)Publication number : 10-320206

(43)Date of publication of application : 04.12.1998

(51)Int.Cl.

G06F 9/44

(21)Application number : 09-232805

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 28.08.1997

(72)Inventor : TAKEWAKI TOSHIKI

(30)Priority

Priority number : 09 60281

Priority date : 14.03.1997

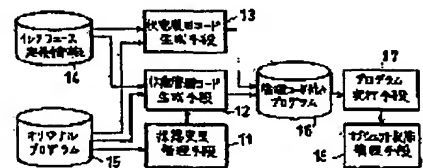
Priority country : JP

(54) METHOD AND DEVICE FOR MANAGING STATE OF OBJECT IN DISTRIBUTED OBJECT SYSTEM,
RECORD MEDIUM FOR PROGRAMMING AND RECORDING THE MANAGING METHOD OR
COMMUNICATION MEDIUM FOR PROPAGATING THE METHOD AS PROGRAM

(57)Abstract:

PROBLEM TO BE SOLVED: To construct a mechanism for automatically recording a change in the state of an object in an external storage device, to make it unnecessary for a programmer to clearly manage the state and to reduce load.

SOLUTION: A state management code generating means 12 for automatically generating a code for holding the state of an object based on previously prepared object interface definition information and a program 15 describing processing provided from the interface and transfers the code to an object state managing means 18 and a state restoration code generating means 13 accesses the object stage in the program describing the processing provided from the interface from the means 18 based on the interface definition information and generates a code for restoring the state.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-320206

(43) 公開日 平成10年(1998)12月4日

(51) Int.Cl.⁶
G 0 6 F 9/44

識別記号
5 3 0

F I
G 0 6 F 9/44

5 3 0 P
5 3 0 M

審査請求 未請求 請求項の数13 O L (全 9 頁)

(21) 出願番号 特願平9-232805

(22) 出願日 平成9年(1997)8月28日

(31) 優先権主張番号 特願平9-60281

(32) 優先日 平9(1997)3月14日

(33) 優先権主張国 日本 (J P)

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 武脇 敏晃

東京都青梅市末広町2丁目9番地 株式会

社東芝青梅工場内

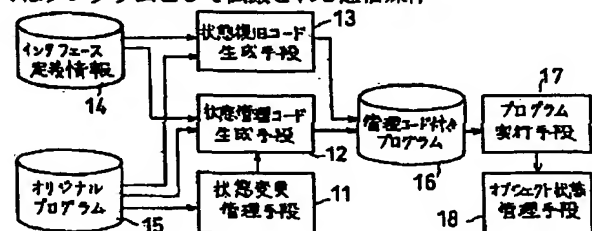
(74) 代理人 弁理士 大胡 典夫 (外1名)

(54) 【発明の名称】 分散オブジェクトシステムにおけるオブジェクトの状態管理方法ならびに装置、及び同方法がプログラムされ記録される記録媒体、もしくはプログラムとして伝搬される通信媒体

(57) 【要約】

【課題】 本発明は、オブジェクトの状態が変化した時、自動的に外部の記憶装置に記録するしくみを構築し、プログラムが明示的に状態を管理することを不要とし負担軽減をはかった分散オブジェクトシステムにおけるオブジェクトの状態管理方法ならびに装置、及び同方法がプログラムされ記憶されるメモリ装置、もしくはプログラムとして伝搬される通信媒体を構築することを主な特徴とする。

【解決手段】 状態管理コード生成手段12は、あらかじめ作成されたオブジェクトのインタフェース定義情報14ならびにそのインタフェースが提供する処理を記述したプログラム15に基づきオブジェクトの状態を保持するコードを自動生成してオブジェクト状態管理手段18に渡し、状態復旧コード生成手段13は、インタフェース定義情報に基づきそのインタフェースが提供する処理を記述したプログラム内のオブジェクト状態をオブジェクト状態管理手段から呼び出し、その状態を復旧するコードを生成する。



【特許請求の範囲】

【請求項1】 ネットワーク上にあるオブジェクトが相互に連携しながら機能する分散オブジェクトシステムにおいて、プログラムの外に存在してオブジェクトの状態を管理するオブジェクト状態管理手段と、

あらかじめ作成されたオブジェクトのインタフェース定義情報ならびにそのインタフェースが提供する処理を記述したプログラムに基づきオブジェクトの状態を保持するコードを自動生成して上記オブジェクト状態管理手段に渡す状態管理コード生成手段とを具備することを特徴とするオブジェクトの状態管理装置。

【請求項2】 上記オブジェクト状態管理手段は、オブジェクトが持つ各属性に対し少なくとも1個の状態を保持する記憶領域を有し、プログラム実行時、この領域に対するアクセスを行うことで各オブジェクトの状態を管理することを特徴とする請求項1記載のオブジェクトの状態管理装置。

【請求項3】 上記オブジェクト状態管理コード生成手段は、オブジェクトのインタフェース定義情報とコード内の状態変更情報に基づいて、各メソッドの処理前に状態を呼び出し、処理後に処理前の状態とを比較して変更された状態のみを上記オブジェクト状態管理手段に出力するコードを生成すること特徴とする請求項1記載のオブジェクト状態管理装置。

【請求項4】 プログラムに基づき各メソッドの中で属性変更の可能性のある状態を静的に解析し、その結果を上記状態管理コード生成手段に渡す状態変更管理手段を更に具備し、上記オブジェクト状態管理コード生成手段は、あらかじめ作成されたオブジェクトのインタフェース定義情報と上記状態変更管理手段から得られる状態変更情報に基づいてそのインタフェースが提供する処理を記述したプログラム内の変更されたオブジェクト状態のみを上記オブジェクト状態管理手段に出力することを特徴とする請求項1記載のオブジェクトの状態管理装置。

【請求項5】 あらかじめ作成されたオブジェクトのインタフェース定義情報に基づきそのインタフェースが提供する処理を記述したプログラム内のオブジェクト状態を上記オブジェクト状態管理手段から呼び出し、その状態を復旧するコードを生成する状態復旧コード生成手段を更に具備することを特徴とする請求項1記載のオブジェクト状態管理装置。

【請求項6】 ネットワーク上にあるオブジェクトが相互に連携しながら機能する分散オブジェクトシステムにおいて、プログラムの外に存在してオブジェクトの状態を管理するオブジェクト状態管理手段と、プログラム実行時に状態の変更が生じたことを検出するオブジェクト状態変更検知手段と、

変更された状態を上記オブジェクト状態管理手段へ出力

するオブジェクト状態更新手段とを具備することを特徴とするオブジェクト状態管理装置。

【請求項7】 オブジェクトの初期化時、上記オブジェクト状態管理手段からオブジェクトの状態を呼び出し状態を復旧するオブジェクト状態復旧手段を更に具備することを特徴とする請求項6記載のオブジェクト状態管理装置。

【請求項8】 データとそれにアクセスするためのメソッドから成るオブジェクトの集まりとしてプログラムが構成され、ネットワーク上に分散配置される分散オブジェクトシステムにおいて、

あらかじめ定められたインタフェース定義情報ならびにそのインタフェースが提供する処理を記述したプログラムから属性リストを作成し、メソッド処理が終了するまでの間にその属性を外部へ通知し状態を保存することを特徴とするオブジェクトの状態管理方法。

【請求項9】 プログラムからメソッドを選択し、上記属性リストに対する復旧コードをメソッド処理の入り口に、管理コードをメソッド処理の出口に挿入することにより外部へ通知し保存することを特徴とする請求項8記載のオブジェクトの状態管理方法。

【請求項10】 プログラムに基づき各メソッドの中で属性変更の可能性のある状態を静的に解析して変更リストを作成し、変更リストに対する復旧コードをメソッド処理の入り口に、管理コードをメソッド処理の出口に挿入することにより外部へ通知し保存することを特徴とする請求項8記載のオブジェクトの状態管理方法。

【請求項11】 データとそれにアクセスするためのメソッドから成るオブジェクトの集まりとしてプログラムが構成され、ネットワーク上に分散配置される分散オブジェクトシステムにおいて、プログラム実行時にオブジェクトの状態変更が生じたことを検出し、変更された状態に更新して外部へ通知することを特徴とするオブジェクト状態管理方法。

【請求項12】 あらかじめ定められたインタフェース定義情報ならびにそのインタフェースが提供する処理を記述したプログラムから属性リストを作成し、メソッド処理が終了するまでの間にその属性を外部へ通知するようにプログラムされ記録、もしくはプログラムとして伝播される、記録媒体、もしくは通信媒体。

【請求項13】 あらかじめ作成されたオブジェクトのインタフェース定義情報に基づきそのインタフェースが提供する処理を記述したプログラム内のオブジェクトの状態を呼び出し、その状態を復旧するコードを生成するようにプログラムされ記録、もしくはプログラムとして伝播される、請求項12記載の記録媒体、もしくは通信媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、分散オブジェクトシステムにおけるオブジェクトの状態管理方法ならびに装置、及び同方法がプログラムされ記録される記録媒体もしくはプログラムとして伝搬される通信媒体に関する。

【0002】

【従来の技術】より複雑なプログラムを作成するためにオブジェクト指向の考え方を採り入れたシステム開発が行われている。オブジェクト指向技術を使用したプログラムの基本は、データとそれにアクセスするためのメソッドからなるオブジェクトであり、オブジェクトの集まりとしてプログラムが構成される。また、ネットワーク技術の発達により、異なる計算機上のオブジェクトを利用する分散オブジェクト技術が注目されてきている。

【0003】分散オブジェクト技術に基づいて作成されるアプリケーションシステムは、異なる計算機で動作するプログラムの集合と見ることができる。また、サービスを提供するオブジェクトは、複数のクライアントプログラムより利用され、内部状態を変化させる。ところが、状態を持ったオブジェクトが、途中で異常終了した場合に、それまでの内部状態は破棄されてしまう状況が生じる。状態を管理するには、ファイルシステムへ状態を保存するなどの処理を明示的に記述する必要がある。

【0004】

【発明が解決しようとする課題】しかしながら、従来オブジェクトの状態を保持する必要がある場合に、明示的にファイルシステムなどに対して状態を記録する処理をプログラマが記述する必要があったため、記録すべきデータの管理が煩雑になることやアプリケーション本来の処理でない記述が増加し、プログラマにかかる負担が大きかった。

【0005】本発明は上記の問題を解決するためになされたものであり、オブジェクトの状態が変化した時、自動的に外部の記憶装置に記録するしくみを構築し、プログラマが明示的に状態を管理することを不要とし負担軽減をはかった、クライアントサーバシステムにおけるオブジェクトの状態管理方法ならびに装置、及び同方法がプログラムされ記憶されるメモリ装置もしくはプログラムとして伝搬される通信媒体を提供することを目的とする。

【0006】

【課題を解決するための手段】本発明は、ネットワーク上にあるオブジェクトが相互に連携しながら機能する分散オブジェクトシステムにおいて、オブジェクト状態管理装置を、プログラムの外に存在してオブジェクトの状態を管理するオブジェクト状態管理手段と、あらかじめ作成されたオブジェクトのインタフェース定義情報ならびにそのインタフェースが提供する処理を記述したプロ

グラムに基づきオブジェクトの状態を保持するコードを自動生成して上記オブジェクト状態管理手段に渡す状態管理コード生成手段で構成したことを特徴とする。

【0007】また、ネットワーク上にあるオブジェクトが相互に連携しながら機能する分散オブジェクトシステムにおいて、オブジェクト状態管理装置を、プログラムの外に存在してオブジェクトの状態を管理するオブジェクト状態管理手段と、プログラム実行時に状態の変更が生じたことを検出するオブジェクト状態変更検知手段と、変更された状態を上記オブジェクト状態管理手段へ出力するオブジェクト状態更新手段で構成したことも特徴とする。

【0008】本発明のオブジェクトの状態管理方法は、データとそれにアクセスするためのメソッドから成るオブジェクトの集まりとしてプログラムが構成され、ネットワーク上に分散配置される分散オブジェクトシステムにおいて、あらかじめ定められたインタフェース定義情報ならびにそのインタフェースが提供する処理を記述したプログラムから属性リストを作成し、メソッド処理が終了するまでの間にその属性を外部へ通知し状態を保存することを特徴とする。また、データとそれにアクセスするためのメソッドから成るオブジェクトの集まりとしてプログラムが構成され、ネットワーク上に分散配置される分散オブジェクトシステムにおいて、プログラム実行時にオブジェクトの状態変更が生じたことを検出し、変更された状態に更新して外部へ通知することも特徴とする。

【0009】更に、本発明の記録媒体もしくは通信媒体は、あらかじめ定められたインタフェース定義情報ならびにそのインタフェースが提供する処理を記述したプログラムから属性リストを作成し、メソッド処理が終了するまでの間にその属性を外部へ通知するようにプログラムされ記憶、もしくはプログラムとして伝搬させることを特徴とする。また、あらかじめ作成されたオブジェクトのインタフェース定義情報に基づきそのインタフェースが提供する処理を記述したプログラム内のオブジェクトの状態を呼び出し、その状態を復旧するコードを生成するようにプログラムされ記憶、もしくはプログラムとして伝搬させることも特徴とする。

【0010】このことにより、オブジェクトの状態が変化した時、自動的に外部の記憶装置に記録するしくみを構築でき、プログラマが明示的に状態を管理することを不要とし負担軽減をはかることができる。

【0011】

【発明の実施の形態】図1は本発明のオブジェクト状態管理装置の機器構成の要部を示すブロック図である。図に示すように、オブジェクト状態管理装置は、機器全体を制御する中央制御装置（CPU）1、CPU1によって実行されるプログラムや各種データを格納する主記憶装置2、主記憶装置2の補助記憶として機能する外部記

憶装置3を少なくとも備え、上述した各装置123はシステムバス4を介して共通接続される。本発明のオブジェクト状態管理装置は、ソフトウェアとしての実施が可能である。従って、フロッピーディスク、ハードディスク等の磁気ディスク、CD-ROM、DVD-ROM、MOなどの光ディスク、および、半導体メモリなどの記憶媒体に格納した形態、あるいは通信媒体を介してソフトウェアとして伝搬させる形態にて提供することが可能である。

【0012】図2に本発明の機能ブロックを示す。図に示される状態変更管理手段11は、オリジナルプログラムに基づいて、各メソッド中で変更される可能性のある状態を静的に解析し、解析結果を状態管理コード生成手段12に渡す。状態管理コード生成手段12は、あらかじめ作成されているオブジェクトのインタフェース定義情報14とそのインタフェースが提供する処理を記述したオリジナルプログラム15に基づいて、状態を保持するためのコードを自動生成する。ここで、状態変更管理手段11から状態変更に関する情報が渡された時には、その情報に基づいて、変更される可能性のある状態だけを保持するコードを生成する。状態管理コード生成手段により生成されたコードは管理コード付きプログラム16として格納される。

【0013】状態復旧コード生成手段13は、あらかじめ作成されているオブジェクトのインタフェース定義情報14とそのインタフェースが提供する処理を記述したオリジナルプログラム15に基づいて、状態を復旧するためのコードを自動生成する。プログラムは、プログラム実行手段17によって実行される。状態管理コード生成手段12や状態復旧コード生成手段13によって、自動生成された状態管理のためのコードも、当該メソッドの実行において処理される。

【0014】オブジェクト状態管理手段18は、インスタンス化されたオブジェクトの各属性に対して、少なくとも1つの状態を保持するための記憶領域を持ち、プログラム実行手段17からこれらの領域に対して書き込みや読み出しを行うことで状態をプログラムの外部で管理するものである。インタフェース定義情報14は、オブジェクト指向技術の標準化団体であるOMG(Object Management Group)で制定したCORBA仕様に含まれるIDL(Interface Definition Language)などでオブジェクトのインタフェースが定義されたものが格納されている。例えば、図3に示すようなインタフェース定義情報があるものとする。インタフェース名I1には2つの属性A1、A2と1つのメソッドM1がある。

【0015】図3に示すインタフェースに対応するオリジナルプログラムの一例を図4に示す。メソッドM1には属性A1に対する書き込みが1箇所だけ存在し、他の属性に対しては操作をしないものとする。set_attribute

(属性名、値)は、属性に値を設定する命令であり、get_attribute(属性名、値)は、属性の値を取り出す命令であるものとする。

【0016】図2を用いて本発明の第1の実施例の動作説明を行う。

【0017】インタフェース定義情報として図3、オリジナルプログラムとして図4に示す例を使用する。状態管理コード生成手段11は、オリジナルプログラムのM1に対して、オブジェクト状態を保持するコードを生成する。この時に、インタフェース定義情報14より、M1がインタフェースI1に属し、属性としてA1、A2があることを得る。これに基づいて、メソッド処理の最終に、属性A1とA2の値をオブジェクト状態管理手段18に書き込むための命令を追加する。図5の破線は、M1に追加されたコードを示している。図中のwrite*status(属性名、値)は、オブジェクト状態管理手段18へ値を書き込むための命令であり、read*status(属性名、値)は、オブジェクト状態管理手段18から値を読み出す命令である。

【0018】次に変形例1について説明する。まず、状態管理コード生成手段12は、オリジナルプログラムのM1に対して、オブジェクト状態を保持するコードを生成する。この時に、インタフェース定義情報14より、M1がインタフェースI1に属し、属性としてA1、A2があることを得る。これに基づいて、メソッド処理の最初に、属性A1、A2の値を一時的な変数(ここでは、S1b、S2b)に保持する命令を追加し、メソッドの最終に、属性A1とA2の値をオブジェクト状態管理手段18に書き込むための命令を追加する。図6の破線はM1に追加されたコードを示している。

【0019】次に変形例2について説明を行う。まず、状態変更管理手段11によって、オリジナルプログラム15に含まれる属性の値の変更操作(set*attribute 命令)を抽出する。図4に示す例では、変更されている属性はA1だけであるため、この情報を状態管理コード生成手段11に渡す。状態管理コード生成手段11では、オリジナルプログラム15のM1に対して、オブジェクト状態を保持するコードを生成する。この時に、状態変更管理手段11から渡された情報に基づいて、メソッドの最終に、属性A1の値をオブジェクト状態管理手段18に書き込むための命令を追加する。図7の破線はM1に追加されたコードを示している。

【0020】上述した変形例との違いは、オブジェクト状態管理コードの生成の違いであり、生成されたコードが実行される時の動作は、そのコードにしたがって振る舞うことになる。プログラムはプログラム実行手段17によって処理され、write*status命令によって、オブジェクト状態管理手段18に当該属性の値が書き込まれる。

【0021】このようにすることで、プログラマが特定

のコードを生成することなく、静的なインタフェース情報に基づいてオブジェクトの状態である属性値を外部に自動的に保持することができる。また、変更点だけを選択的に管理することによって、効率的に状態保持をおこなうことができる。更に、静的なプログラム解析により状態の変更点だけを選択的に管理するため、効率的に状態保持をおこなうこともできるようになる。

【0022】次にオブジェクト状態の復旧に関する動作についてを説明する。ここでは、第1の実施例に従う方法で状態が管理されているものとして説明を行う。まず、状態復旧コード生成手段13は、オリジナルプログラム15のメソッドM1に対して、オブジェクト状態を復旧するコードを生成する。この時に、インタフェース定義情報14から、M1がインタフェースI1に属し、属性としてA1、A2があることを得る。これに基づいて、メソッド処理の最初に、属性A1とA2の値をオブジェクト状態管理手段18から読み込むための命令を追加する。図8に示す破線は、M1に追加されたコードを示している。もし、当該属性に対する値が設定されていないときはデフォルト値を返すものとする。

【0023】上述した動作は、図10乃至図11にフローチャートとして示されている。

【0024】いずれもオリジナルプログラム15からクラス(モジュール)を選択し、当該クラスのインタフェース定義情報14から必要なインタフェース情報を抽出して属性リストを作成する。そして、オリジナルプログラム15からメソッドを選択する。そして属性リストに対する復旧コードをメソッドの入入口、管理コードをメソッドの出口に挿入する動作をメソッド単位で行う。

【0025】尚、変形例2では、オリジナルプログラム15からメソッドを選択した後、そのメソッド内での属性変更可能性から変更リストを作成し、属性リストではなく、変更リストに対する復旧コードの挿入処理を行う点が図10に示す変形例1の動作と異なる。これらの動作により、オブジェクトが異常終了した後に、以前のオブジェクト状態に自動的に復旧させることができるようになる。

【0026】以下、本発明の他の実施例について説明する。オブジェクト状態管理装置の機器構成は、図1に示すように第1の実施例と同じである。

【0027】図9にこの実施例の機能ブロック図を示す。図9に示されるプログラム実行手段91は、プログラムの実行を行う部分である。状態変更・更新手段92によって、プログラム実行手段91によってなされる属性の変更を検出し、この値をオブジェクト状態保持手段93に送る。オブジェクト状態管理手段93は、インスタンス化されたオブジェクトの各アトリビュートに対して、少なくとも1つの状態を保持するための領域を持ち、状態検出・更新手段92から送られてくる情報にもとづいて、これらの領域に対して値の書き込みを行うこ

とで状態をプログラムの外部で管理するものである。状態復旧手段94は、オブジェクトの状態をオブジェクト保持手段から呼び出し、当該オブジェクトの属性値を設定することで、オブジェクトの状態を復旧する。

【0028】同実施例の動作について説明する。動作の説明のためにプログラムとして図4に示す例を引用する。図4に示すプログラムは、属性A1への値の書き込みが行われるだけである。プログラム実行手段91において、図4のプログラム実行中に、set*attribute 命令に達したとき、状態変更・更新手段92に対して、属性の変更が行われたことを通知する。この通知に基づいて、状態変更・更新手段92は、オブジェクト状態保持手段93に対して属性A1が5に変更されたことを登録する。オブジェクト状態保持手段93では、属性に対する領域がないときには、新たに領域を確保する。既に存在するときは、上書きにより値を更新する。

【0029】このオブジェクトが何らかの原因により異常終了した場合に、次にオブジェクトを利用する要求が起こったときに、オブジェクトの再起動が発生する。ORBのようなオブジェクト管理機構があれば、プログラムの再起動は自動的に行うことが可能であるが、状態は初期化される。状態復旧手段94は、プログラムが再起動されるときに、オブジェクト状態保持手段93から当該オブジェクトに含まれる属性に関する値を読みだし、オブジェクト内に設定する。もし、属性に対する値の保持領域がないときには、オブジェクトの属性値としてデフォルト値を設定する。

【0030】これらの動作により、プログラムに実行時に動的に変更された状態を検出し、プログラムの外部に記録することができる。また、オブジェクトが異常終了した後に、以前のオブジェクト状態に自動的に復旧させることができる。更にソースコードに手を加えることなくオブジェクト状態の記録を行い、かつオブジェクト状態の復旧を行うことが可能になる。

【0031】尚、上述した手法は、コンピュータに実行させることのできるプログラム情報として、例えば、磁気ディスク装置(フロッピーディスク、ハードディスク等)、光ディスク装置(CD-ROM、DVD等)半導体メモリ等の記録媒体に書き込んで各種装置に適用したり、通信媒体を介して伝送して各種装置に適用することは勿論可能である。この装置を実現するコンピュータは、記録媒体に記録されたプログラム情報を読み取り、このプログラム情報によって動作が制御されることにより上述した処理を実行する。

【0032】

【発明の効果】以上説明のように本発明は、オブジェクトの状態が変化した時、自動的に外部の記憶装置に記録するしくみを構築し、プログラマが明示的に状態を管理することを不要とし負担軽減をはかったものであり、更に以下に列挙する効果が得られる。

【0033】(1) プログラムが特定のコードを生成することなく、静的なインタフェース情報に基づいてオブジェクトの状態である属性値を外部に自動的に保持することができる。

【0034】(2) 変更点だけを選択的に管理するため、効率的に状態保持をおこなうことができ、動的に状態を変更を選択する方法である。

【0035】(3) 静的なプログラム解析により状態の変更点だけを選択的に管理するため、効率的に状態保持をおこなうことができる。

【0036】(4) オブジェクトが異常終了した後に、以前のオブジェクト状態に自動的に復旧させることができる。

【0037】(5) プログラムに実行時に動的に変更された状態を検出し、プログラムの外部に記録することができる。ソースコードの手を加えることなく、オブジェクト状態の記録を行う。

【0038】(6) オブジェクトが異常終了した後に、以前のオブジェクト状態に自動的に復旧させることができる。ソースコードの手を加えることなく、オブジェクト状態の復旧を行う。

【図面の簡単な説明】

【図1】本発明のオブジェクト状態管理装置の機器構成の要部を示すブロック図

【図2】本発明実施例の機能ブロックを示す図。

【図3】本発明にて用いられるインタフェース定義情報

の一例を示す図。

【図4】図3に示すインタフェースに対応するオリジナルプログラムの一例を示す図。

【図5】本発明実施例の動作を説明するために記述したプログラムの例を示す図。

【図6】本発明実施例の動作を説明するために記述したプログラムの例を示す図。

【図7】本発明実施例の動作を説明するために記述したプログラムの例を示す図。

【図8】本発明実施例の動作を説明するために記述したプログラムの例を示す図。

【図9】本発明実施例の動作を説明するために引用したフローチャート。

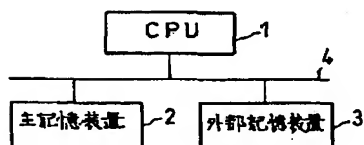
【図10】本発明実施例の動作を説明するために引用したフローチャート。

【図11】本発明の他の実施例による機能ブロックを示す図。

【符号の説明】

1…中央処理装置(CPU)、2…主記憶装置、3…外部記憶装置、4…システムバス、11…状態変更管理手段、12…状態管理コード生成手段、13…状態復旧コード生成手段、14…インタフェース定義情報、15…オリジナルプログラム、16…管理コード付きプログラム、17、91…プログラム実行手段、18…オブジェクト状態管理手段、92…状態変更検出・更新手段、93…オブジェクト状態保持手段、94…状態復旧手段。

【図1】



【図3】

```

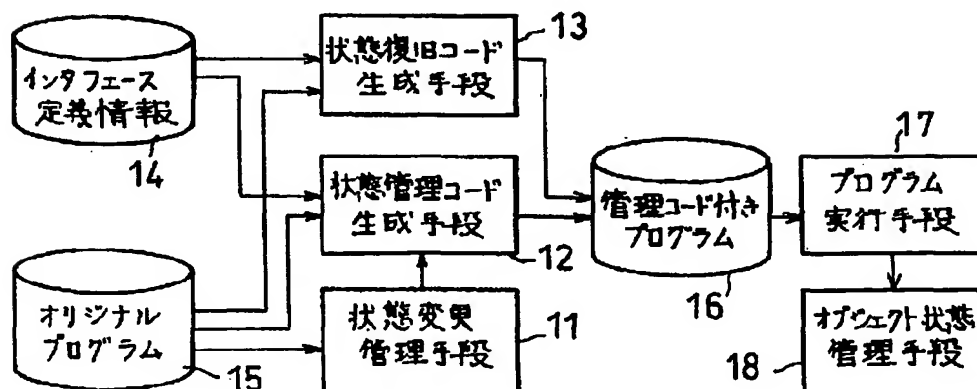
interface I{
  attribute long A1;
  attribute long A2;
  void MI (-);
};
  
```

【図4】

```

MI (-){
  ...
  set_attribute (A1,5);
  ...
}
  
```

【図2】



【図5】

```

MI(...){
    ...
    set_attribute(A1,5)
    ...
    get_attribute(A1,S1a)
    write_status (A1,S1a)
    get_attribute (A2,S2a)
    write_status (A2,S2a)
}

```

【図6】

```

MI(...){
    ...
    get_attribute(S1b)
    get_attribute(S2b)
    ...
    set_attribute(A1,5)
    ...
    get_attribute(A1,S1a)
    if(S1b != S1a)
    { write_status(A1,S1a) }
    get_attribute(A2,S2a)
    if(S2b != S2a)
    { write_status(A2,S2a) }
}

```

【図7】

```

MI(...){
    ...
    set_attribute(A1,5)
    ...
    get_attribute(A1,S1a)
    write_status(A1,S1a)
}

```

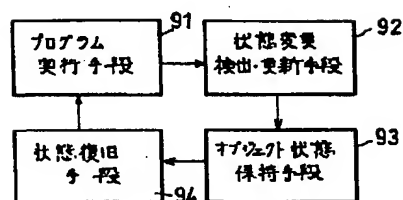
【図8】

```

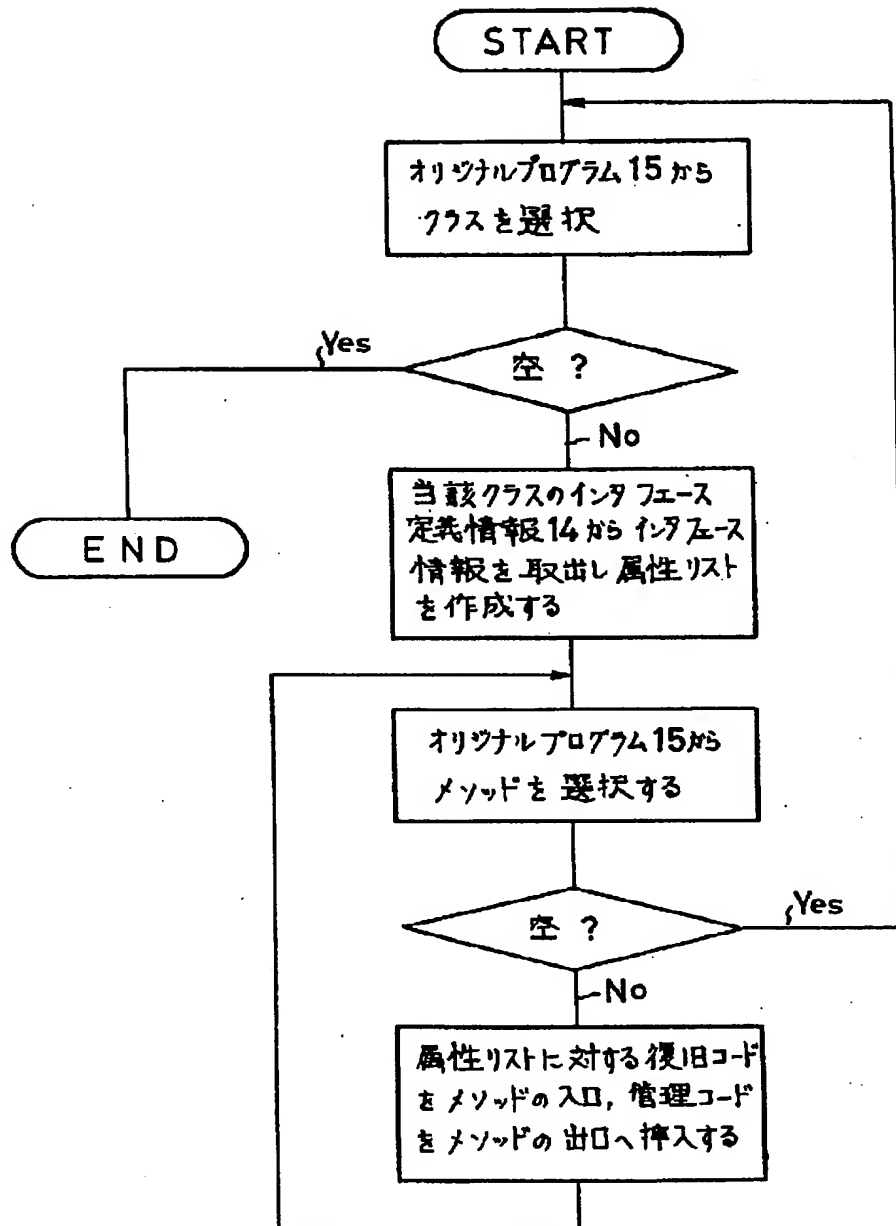
MI(...){
    ...
    read_status(A1,S1b)
    set_attribute(A1,S1b)
    read_status(A2,S2b)
    set_attribute(A2,S2b)
    ...
    set_attribute(A1,5)
    ...
}

```

【図9】



【図10】



【図11】

